

IN THE CLAIMS

Claims 1 – 9 (Cancelled)

10. (Currently amended) A[[The]] method [[of claim 9]] for balancing multiple memory buffer sizes comprising:

building a buffer steal history queue comprising one or more buffer steal history records, wherein each of the one or more buffer steal history records includes a unique identifier for identifying one or more cached data stored in a secondary memory location, wherein the buffer steal history record further comprises a sequence number [[, the method further comprising:]]:

receiving a request to store data in a buffer;

comparing the request which has been received to each of the unique identifiers of each of the one or more buffer steal history records;

reaccessing the cached data from the secondary memory when the data in the request matches one of the unique identifiers of a buffer steal history record;

computing a difference between the sequence number of the buffer steal history record and a current sequence number for a corresponding buffer size subgroup;

comparing the difference to the corresponding buffer size subgroup; and

reaccessing the corresponding buffer size subgroup for the cached data when the corresponding buffer size subgroup is at least equal to the difference.

11. (Previously Presented) The method of claim 10, further comprising:

rereading the cached data from the secondary memory when the corresponding buffer size subgroup is smaller than the difference.

12. (Previously Presented) The method of claim 11, wherein the difference comprises the number of buffers stolen from the corresponding buffer size subgroup between a prior buffer steal and a current requested reaccess.

13. (Currently Amended) The method of claim 11, further comprising:
for each of a plurality of subgroups, wherein each subgroup comprises a plurality of
[similarly sized]buffers whose size is substantially equal, calculating a priority for the
subgroup by:

accumulating an average difference for all buffers stolen and reaccessed
from the subgroup;
determining an importance of reaccesses in the subgroup; and
identifying a buffer steal target in response to a request for a buffer by
comparing the priority of each subgroup, wherein the buffer steal target is a
buffer having [[the]]a lowest priority.

Claims 14-29 (Cancelled)

30. (New) A computer program product for balancing multiple memory buffer sizes, the
computer program product comprising instructions for:

building a buffer steal history queue comprising one or more buffer steal history
records, wherein each of the one or more buffer steal history records includes a unique
identifier for identifying one or more cached data stored in a secondary memory
location, wherein the buffer steal history record further comprises a sequence number;
receiving a request to store data in a buffer;
comparing the request which has been received to each of the unique identifiers
of each of the one or more buffer steal history records;
reaccessing the cached data from the secondary memory when the data in the
request matches one of the unique identifiers of a buffer steal history record;
computing a difference between the sequence number of the buffer steal history
record and a current sequence number for a corresponding buffer size subgroup;
comparing the difference to the corresponding buffer size subgroup; and

reaccessing the corresponding buffer size subgroup for the cached data when the corresponding buffer size subgroup is at least equal to the difference.

31. (New) The computer program product of claim 30, further comprising instructions for:

rereading the cached data from the secondary memory when the corresponding buffer size subgroup is smaller than the difference.

32. (New) The computer program product of claim 31, wherein the difference comprises the number of buffers stolen from the corresponding buffer size subgroup between a prior buffer steal and a current requested reaccess.

33. (New) The computer program product of claim 31, further comprising instructions for each of a plurality of subgroups, wherein each subgroup comprises a plurality of buffers whose size is substantially equal, calculating a priority for the subgroup by:

accumulating an average difference for all buffers stolen and reaccessed from the subgroup;

determining an importance of reaccesses in the subgroup; and

identifying a buffer steal target in response to a request for a buffer by comparing the priority of each subgroup, wherein the buffer steal target is a buffer having a lowest priority.

34. (New) A system for balancing multiple memory buffer sizes comprising:

a buffer steal history queue comprising one or more buffer steal history records, wherein each of the one or more buffer steal history records includes a unique identifier for identifying one or more cached data stored in a secondary memory location, wherein the buffer steal history record further comprises a sequence number;

a controller including a memory manager, wherein the memory manager is adapted to:

receive a request to store data in a buffer;
compare the request which has been received to each of the unique identifiers of each of the one or more buffer steal history records;
reaccess the cached data from the secondary memory when the data in the request matches one of the unique identifiers of a buffer steal history record;
compute a difference between the sequence number of the buffer steal history record and a current sequence number for a corresponding buffer size subgroup;
compare the difference to the corresponding buffer size subgroup; and
reaccess the corresponding buffer size subgroup for the cached data when the corresponding buffer size subgroup is at least equal to the difference.

35. (New) The system of claim 34, wherein the controller is further adapted to:

reread the cached data from the secondary memory when the corresponding buffer size subgroup is smaller than the difference.

36. (New) The system of claim 35, wherein the difference comprises the number of buffers stolen from the corresponding buffer size subgroup between a prior buffer steal and a current requested reaccess.

37. (New) The system of claim 34, further comprising instructions for each of a plurality of subgroups, wherein each subgroup comprises a plurality of buffers whose size is substantially equal, calculating a priority for the subgroup by:

accumulating an average difference for all buffers stolen and reaccessed from the subgroup;
determining an importance of reaccesses in the subgroup; and
identifying a buffer steal target in response to a request for a buffer by comparing the priority of each subgroup, wherein the buffer steal target is a buffer having a lowest priority.